APPENDIX A

```
     #define MAX_DIV_DEPTH 23
 5   static UINT32 divReload[MAX_DIV_DEPTH];
     static UINT32 divCurrent[MAX_DIV_DEPTH];
     static UINT32 divDepth;

     /* -------------------------------------------------
10   * initialisations : compute the partial quotients a(i)
     */
     static void divR (UINT32 v1, UINT32 v2)
     {
         assert(divDepth < MAX_DIV_DEPTH);
15       divReload[divDepth++] = v1 / v2;
         v2 = v1 % v2;
         if (v1 != 0 && v2 != 0) divR (v2, v1);
     }


20   /* -------------------------------------------------
     * initialization entry point
     */
     void GRDASet (UINT32 P, UINT32 Q)
     {
25       assert(P>=Q);
         if (P == 0 || Q == 0)
         {
           divDepth = 1; divReload[0] = 0;
         }
30       else
         {
           divDepth = 0; divR (P, Q);
           for (int i = 0; i < divDepth; i++) divCurrent[i] =
     (divReload[i] + 1)/2;
35       } /* end of if-else(v1) */
     }


     /* -------------------------------------------------
     * Get successive values g(i)
40   */
     UINT32 GRDAGet (void)
     {
         /* process the first coefficients 0 and 1 */
         if (divDepth < 2 || --divCurrent[1] != 0) return
45   divReload[0];
```

```
       divCurrent[1] = divReload[1];
       /* process the remaining coefficients */
       int i = 1;
       while (++i < divDepth)
5      {
           if (--divCurrent[i] != 0) break;
           divCurrent[i] = divReload[i];
           divCurrent[i - 1]++;
       }
10     return divReload[0] + 1;
    }
```